(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平9－311821

(43)公開日　平成９年(1997)12月２日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　12/14	３１０		Ｇ０６Ｆ　12/14	３１０Ｋ	
				３１０Ａ	

審査請求　未請求　請求項の数６　ＯＬ　（全 18 頁）

(21)出願番号　特願平8－128635

(22)出願日　平成８年(1996)５月23日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　古林　三郎
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

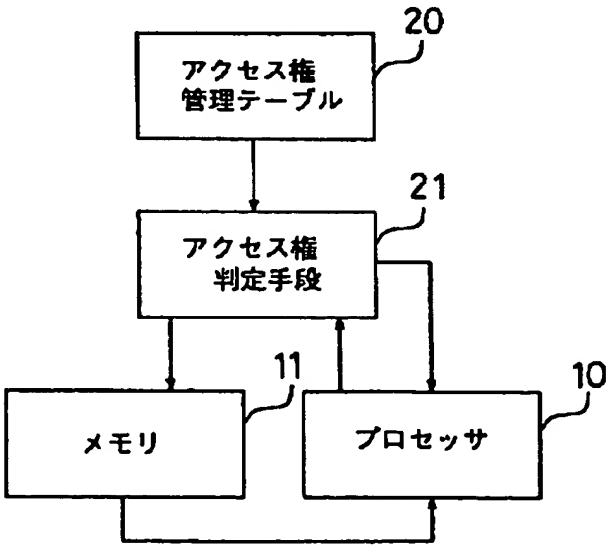
(74)代理人　弁理士　田澤　博昭　（外２名）

(54)【発明の名称】　記憶データ保護装置

(57)【要約】

【課題】　記憶データの保護が、オペレーティングシステムにおける制御の基本単位であるプロセス間でしか行えなかった。

【解決手段】　メモリ１１上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ１１上のモジュールのアドレス範囲を格納しているアクセス権管理テーブル２０と、プロセッサ１０から要求されたアドレスがアクセス権管理テーブル２０で定義されたデータセグメントである場合にそのアクセス権管理テーブル２０を用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段２１とを備えた。



1

【特許請求の範囲】

【請求項１】　プロセッサおよびメモリを有する電子計算機における単一のプログラム内の各種機能モジュール間で記憶データの保護を行う記憶データ保護装置において、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ上のモジュールのアドレス範囲を格納するアクセス権管理テーブルと、上記プロセッサから要求されたアドレスが上記アクセス権管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段とを備えたことを特徴とする記憶データ保護装置。

【請求項２】　アクセス権管理テーブルは、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールを上記モジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えたことを特徴とする請求項１記載の記憶データ保護装置。

【請求項３】　アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えたことを特徴とする請求項２記載の記憶データ保護装置。

【請求項４】　アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールを上記データセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えたことを特徴とする請求項３記載の記憶データ保護装置。

【請求項５】　アクセス権判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、そのトラップ検出手段がトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えたことを特徴とする請求項４記載の記憶データ保護装置。

【請求項６】　各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じて上記フラグを検査し、切り離しが可能であれば上記切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段とを備えたことを特徴とする請求項５記載の記憶データ保護装置。

2

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、プロセッサおよびメモリを有する電子計算機において、単一のプログラム内の各種機能モジュール間で記憶データの保護を行う記憶データ保護装置に関するものである。

【０００２】

【従来の技術】図２１は例えば特開平４－１９９３４１号公報に示された従来の記憶データ保護装置を示すブロック図であり、図において、１０はプロセッサ、１１はメモリ、１２はメモリ１１上のアドレス範囲で定義される各種データセグメント毎にアクセス可能なプロセスを示すメモリアクセスキーを格納するアクセス権管理テーブルである。１３は現在走行するプロセスのメモリアクセスキーを保持するアクセスキーレジスタ、１４はアクセス権管理テーブル１２とアクセスキーレジスタ１３を用いて現在走行するプロセスがプロセッサ１０から要求されたアドレスに対応するデータセグメントにアクセス可能であるかどうかを判定するアクセス権判定手段である。アクセスキーレジスタ１３には、プロセスが走行を開始する時に対応するメモリアクセスキーが設定される。

【０００３】図２２はアクセス権管理テーブルの構造を示す説明図であり、図において、１５はアクセス権管理テーブル１２のインデックスとしてのデータセグメントインデックス、１６はデータセグメント開始アドレス、１７はデータセグメントサイズ、１８はデータセグメントインデックス１５で示されるデータセグメントに対してアクセス可能なプロセスのメモリアクセスキーである。

【０００４】次に動作について説明する。図２３は従来の記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、この図２３を用いて説明する。まず、ステップＳＴ２３１でプロセッサ１０がアクセスしたいアドレスをアクセス権判定手段１４に送ると、アクセス権判定手段１４は、ステップＳＴ２３２で送られたアドレスをもとにアクセス権管理テーブル１２を参照し、アドレスに対応したメモリアクセスキー１８を取得する。

【０００５】次に、ステップＳＴ２３３で、取得したメモリアクセスキー１８をアクセスキーレジスタ１３の内容と比較し、現在走行しているプロセスに対してアクセスが許可されているかどうかの判定を行う。アクセスが許可されていると判定した場合は、ステップＳＴ２３４に進み、アドレスをメモリ１１に送る。そして、ステップＳＴ２３５で、メモリ１１はアドレスに対応したメモリ１１の内容をプロセッサ１０に送る。ステップＳＴ２３３にてアクセスが許可されていないと判定した場合

は、ステップＳＴ２３６に進み、プロセッサ１０に対して不当アクセス要求であることを通知する。

【０００６】

【発明が解決しようとする課題】従来の記憶データ保護装置は以上のように構成されているので、記憶データの保護が、オペレーティングシステムにおける制御の基本単位であるプロセス間でしか行うことができなかった。プロセス、即ちプログラムは複数の機能モジュールの集合体である。例えば、複数の開発者が作成した異なる機能モジュールを組み上げてできるプログラムにおいて、ある機能モジュールが全く関係のない他の機能モジュールのデータを破壊してしまったような場合、従来技術による記憶データ保護装置は効力を持つことができない。この場合、不当なアクセスにも拘らずその不当なアクセスを検出することができないため、プログラムの暴走を引き起こしてしまうなどの課題があった。

【０００７】この発明は上記のような課題を解決するためになされたもので、１つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる記憶データ保護装置を得ることを目的とする。

【０００８】

【課題を解決するための手段】請求項１記載の発明に係る記憶データ保護装置は、メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ上のモジュールのアドレス範囲を格納するアクセス権管理テーブルと、プロセッサから要求されたアドレスが上記アクセス権管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段とを備えたものである。

【０００９】請求項２記載の発明に係る記憶データ保護装置は、アクセス権管理テーブルに、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールを上記モジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えたものである。

【００１０】請求項３記載の発明に係る記憶データ保護装置は、アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えたものである。

【００１１】請求項４記載の発明に係る記憶データ保護装置は、アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを

登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールを上記データセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えたものである。

【００１２】請求項５記載の発明に係る記憶データ保護装置は、アクセス権判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、そのトラップ検出手段がトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えたものである。

【００１３】請求項６記載の発明に係る記憶データ保護装置は、各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じて上記フラグを検査し、切り離しが可能であれば上記切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段とを備えたものである。

【００１４】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態１．図１はこの発明の実施の形態１による記憶データ保護装置を示すブロック図であり、図において、１０はプロセッサ、１１はメモリ、２０はメモリ１１上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ１１上のモジュールのアドレス範囲を格納しているアクセス権管理テーブル、２１はプロセッサ１０から要求されたアドレスがアクセス権管理テーブル２０で定義されたデータセグメントである場合にアクセス権管理テーブル２０を用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段である。

【００１５】図２はアクセス権管理テーブルの構造を示す説明図であり、図において、１５はアクセス権管理テーブル２０のインデックスとしてのデータセグメントインデックス、１６はデータセグメント開始アドレス、１７はデータセグメントサイズである。２２は各データセグメントインデックス１５に対応するデータセグメントについてのアクセス可能モジュール開始アドレス、２３はアクセス可能モジュールサイズである。

【００１６】次に動作について説明する。図３はこの実施の形態１による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、以下、この図３を参照しながら説明する。まず、ステップＳＴ３１でプロセッサ１０がアクセスしたいアドレスをアクセス権判定手段２１に送ると、アクセス権判定手段２１

は、ステップＳＴ３２で送られたアドレスをもとにアクセス権管理テーブル２０のデータセグメント開始アドレス１６とデータセグメントサイズ１７を参照し、送られたアドレスに対応するデータセグメントインデックス１５を取得する。

【００１７】次に、ステップＳＴ３３で、現在のプログラムカウンタが取得したモジュールと、送られたアドレスに対応するデータセグメントインデックス１５のアクセス可能モジュール開始アドレス２２及びアクセス可能モジュールサイズ２３から、現在のプログラムカウンタが取得したモジュールがアクセス可能モジュールの範囲内であるかどうかを検査して、アクセスが許可されているかどうか判定を行う。アクセスが許可されていると判定した場合は、ステップＳＴ３４に進み、アドレスをメモリ１１に送る。そして、ステップＳＴ３５で、メモリ１１はアドレスに対応したメモリ１１の内容をプロセッサ１０に送る。ステップＳＴ３３にてアクセスが許可されていないと判定した場合は、ステップＳＴ３６に進み、プロセッサ１０に対して不当アクセス要求であることを通知する。

【００１８】以上のように、この実施の形態１によれば、現在のプログラムカウンタをもとにアクセス権管理テーブル２０を参照して、モジュール単位で記憶データの保護を行うことができるため、１つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる効果がある。

【００１９】実施の形態２．図４はこの発明の実施の形態２による記憶データ保護装置を示すブロック図であり、図において、３０はメモリ１１上のアドレス範囲で各種モジュールを定義するモジュール管理テーブル、３１はメモリ１１上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールをモジュール管理テーブル３０上のインデックスで複数登録されたデータセグメント管理テーブルである。

【００２０】図５はモジュール管理テーブルの構造を示す説明図であり、図において、３２はモジュール管理テーブル３０のインデックスとしてのモジュールインデックス、３３はモジュール開始アドレス、３４はモジュールサイズである。図６はデータセグメント管理テーブルの構造を示す説明図であり、図において、３５はデータセグメントインデックス１５に対応するデータセグメントに対してアクセスが許可されたモジュールをモジュールインデックス３２で示すアクセス可能モジュールインデックスである。なお、その他の構成については上記実施の形態１と同一なのでその重複する説明を省略する。

【００２１】次に動作について説明する。図７はこの実施の形態２による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、以下、この図７を参照しながら説明する。まず、ステップＳＴ７１でプロセッサ１０がアクセスしたいアドレスをアクセス権判定手段２１に送ると、アクセス権判定手段２１は、ステップＳＴ７２では送られたアドレスをもとにデータセグメント管理テーブル３１のデータセグメント開始アドレス１６とデータセグメントサイズ１７を参照し、送られたアドレスに対応するデータセグメントインデックス１５を取得する。

【００２２】次に、ステップＳＴ７３で、取得したデータセグメントインデックス１５に対応するアクセス可能モジュールインデックス３５を順次取得する。そして、ステップＳＴ７４にて、モジュール管理テーブル３０において、現在のプログラムカウンタが取得したモジュールと、アクセス可能モジュールインデックス３５によって順次取得されたモジュールインデックス３２のモジュール開始アドレス３３及びモジュールサイズ３４から、現在のプログラムカウンタが取得したモジュールがアクセス可能モジュールの範囲内であるかどうかを検査して、アクセスが許可されているかどうかの判定を行う。アクセスが許可されていると判定した場合は、ステップＳＴ７５に進み、アドレスをメモリ１１に送る。そして、ステップＳＴ７６で、メモリ１１はアドレスに対応したメモリ１１の内容をプロセッサ１０に送る。

【００２３】ステップＳＴ７４にてアクセスが許可されていないと判定した場合は、ステップＳＴ７７に進み、全てのアクセス可能モジュールインデックス３５について検査が終了していなかった場合、再びステップＳＴ７３に戻ってアクセス権判定を行う。ステップＳＴ７７にて全てのアクセス可能モジュールインデックス３５について検査が終了していた場合、ステップＳＴ７８に進み、プロセッサ１０に対して不当アクセス要求であることを通知する。

【００２４】以上のように、この実施の形態２によれば、アクセス権判定手段２１がアクセスしたデータセグメントについて、アクセス可能モジュールインデックス３５として登録されているすべてのモジュールを検査するため、複数のモジュール間で１つのデータセグメントを共有することができる効果がある。

【００２５】実施の形態３．図８はこの発明の実施の形態３による記憶データ保護装置を示すブロック図であり、図において、４０はアクセス権管理テーブル２０を更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブル２０の拡張領域を確保するアクセス権管理テーブル更新手段である。なお、その他の構成については上記実施の形態２と同一なのでその重複する説明を省略する。

【００２６】次に動作について説明する。図９はこの実施の形態３による記憶データ保護装置におけるアクセス権管理テーブル更新処理の流れを示すフローチャートであり、以下、この図９を参照しながら説明する。アクセス権管理テーブル更新手段４０は、まず、ステップＳＴ９１で、初期設定であった場合にステップＳＴ９２に進

み、アクセス権管理テーブル２０用の領域を確保する。次にステップＳＴ９３で、モジュール管理テーブル３０におけるモジュールインデックス３２が０のエントリにアクセス権管理テーブル更新モジュール４０用のエントリを作成する。そしてステップＳＴ９４で、データセグメント管理テーブル３１におけるデータセグメントインデックス１５が０のエントリにアクセス権管理テーブル２０用のエントリを作成する。

【００２７】次にステップＳＴ９５にて、アクセス権管理テーブル２０を更新するための十分な空き領域があるかないかを検査し、空き領域が十分あると判定された場合、ステップＳＴ９６に進み、アクセス権判定手段２１を用いたメモリアクセスにより要求された更新処理を行う。ステップＳＴ９１にて初期設定でなかった場合はステップＳＴ９５に進み、アクセス権管理テーブル２０を更新するための十分な空き領域があるかないかを検査する。ステップＳＴ９５にて空き領域が不足と判定された場合、ステップＳＴ９７に進み、アクセス権管理テーブル２０用の拡張領域を確保する。次にステップＳＴ９８で、データセグメント管理テーブル３１の空きエントリにアクセス権管理テーブル２０用の追加エントリを作成する。そして、ステップＳＴ９６で、アクセス権判定手段２１を用いたメモリアクセスにより要求された更新処理を行う。

【００２８】以上のように、この実施の形態３によれば、アクセス権管理テーブル２０を更新できるのはアクセス権管理テーブル更新手段４０だけとなるため、アクセス権管理テーブル２０を誤操作による破壊から守り、安全に更新することができる効果がある。

【００２９】実施の形態４．図１０はこの発明の実施の形態４による記憶データ保護装置を示すブロック図であり、図において、５０はモジュール管理テーブル３０にモジュールを登録するモジュールインストール手段、５１はメモリの獲得または解放に合わせてデータセグメント管理テーブル３１にデータセグメントを登録したり削除したりするメモリ獲得解放手段、５２はアクセス許可を申請したモジュールをデータセグメント管理テーブル３１のアクセス可能モジュールインデックス３５に追加するデータアクセス宣言手段である。なお、その他の構成については上記実施の形態３と同一なのでその重複する説明を省略する。

【００３０】次に動作について説明する。図１１はこの実施の形態４による記憶データ保護装置におけるモジュールインストール処理の流れを示すフローチャートであり、以下、この図１１を参照しながら説明する。モジュールインストール手段５０は、まず、ステップＳＴ１１１で、要求されたモジュールをインストールする。そして、ステップＳＴ１１２で、アクセス権管理テーブル更新手段４０を用いてモジュール管理テーブル３０にインストールしたモジュールのエントリを追加する。

【００３１】図１２はメモリ獲得解放処理の流れを示すフローチャートであり、以下、この図１２を参照しながら説明する。メモリ獲得解放手段５１は、まず、ステップＳＴ１２１で、要求がメモリ獲得か解放かを判別し、獲得要求の場合にステップＳＴ１２２に進み、アクセス権管理テーブル更新手段４０を用いてデータセグメント管理テーブル３１に獲得したデータセグメント用のエントリを作成する。次にステップＳＴ１２３で、モジュール管理テーブル３０と現在のプログラムカウンタから要求元モジュールに対応するモジュールインデックス３２を取得する。

【００３２】そして、ステップＳＴ１２４で、アクセス権管理テーブル更新手段４０を用いて取得したモジュールインデックス３２をアクセス可能モジュールインデックス３５としてデータセグメント管理テーブル３１に登録する。ステップＳＴ１２１にて、解放要求であった場合、ステップＳＴ１２５に進み、アクセス権管理テーブル更新手段４０を用いてデータセグメント管理テーブル３１から対応するデータセグメントのエントリを削除する。

【００３３】図１３はデータアクセス宣言処理の流れを示すフローチャートであり、以下、この図１３を参照しながら説明する。データアクセス宣言手段５２は、まず、ステップＳＴ１３１で、データセグメント管理テーブル３１から宣言されたアドレスに対応するデータセグメントを特定する。次にステップＳＴ１３２で、モジュール管理テーブル３０と現在のプログラムカウンタから要求元モジュールに対応するモジュールインデックス３２を取得する。

【００３４】そして、ステップＳＴ１３３で、取得したモジュールインデックス３２が、アクセス宣言されたデータセグメントに対して既にアクセス許可されているかどうかを検査し、まだ許可されていなかった場合は、ステップＳＴ１３４に進み、アクセス権管理テーブル更新手段４０を用いて取得したモジュールインデックス３２をアクセス可能モジュールインデックス３５としてデータセグメント管理テーブル３１に登録する。ステップＳＴ１３３にて既に許可されていた場合は、そのまま処理を終了する。

【００３５】以上のように、この実施の形態４によれば、モジュールインストール手段５０がモジュールインストール時にモジュール管理テーブル３０を作成し、メモリ獲得解放手段５１がデータセグメントの生成または削除に合わせてデータセグメント管理テーブル３１の更新を行い、データアクセス宣言手段５２がモジュールが自発的に宣言した時にアクセス可能モジュールインデックス３５として上記モジュールをデータセグメント管理テーブル３１に登録するため、動的に生成及び削除されるようなデータに対するアクセス権管理テーブル２０の更新処理を自動的に行うことができる効果がある。

【００３６】実施の形態５．図１４はこの発明の実施の形態５による記憶データ保護装置を示すブロック図であり、図において、６０は二次記憶装置、６１はアクセス権判定手段２１によって検出された不当アクセスによるプロセッサ１０からの通知を検出するトラップ検出手段、６２はトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブル２０の内容とプログラムカウンタとアクセスアドレスを二次記憶装置６０に記録する不当アクセス内容記録手段である。なお、その他の構成については上記実施の形態４と同一なのでその重複する説明を省略する。

【００３７】次に動作について説明する。図１５はこの実施の形態５による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートであり、以下、この図１５を参照しながら説明する。トラップ検出手段６１は、まず、ステップＳＴ１５１で、プロセッサ１０からの不当アクセスの通知をトラップとして検出する。次にステップＳＴ１５２で、不当アクセス内容記録手段６２を用いて不当アクセスの内容を二次記憶装置６０上に記録する。そして、ステップＳＴ１５３で、アクセス要求元に対して不当アクセスを通知する。

【００３８】図１６は不当アクセス内容記録処理の流れを示すフローチャートであり、以下、この図１６を参照しながら説明する。不当アクセス内容記録手段６２は、まず、ステップＳＴ１６１で、現在時刻を取得する。次にステップＳＴ１６２で、二次記憶装置６０上に取得した現在時刻をタグとする領域を確保する。そして、ステップＳＴ１６３で、確保した不当アクセス内容記録領域にアクセス権管理テーブル２０の内容を記録し、続くステップＳＴ１６４で、上記領域にトラップ検出時のプログラムカウンタとアクセスアドレスを記録する。

【００３９】以上のように、この実施の形態５によれば、プロセッサ１０からの不当アクセスの通知を検出したトラップ検出手段６１が、不当アクセス内容記録手段６２を用いて不当アクセスの内容を二次記憶装置６０に記録するため、不当アクセスの内容を確実に且つ容易に知ることができる効果がある。

【００４０】実施の形態６．図１７はこの発明の実施の形態６による記憶データ保護装置を示すブロック図であり、図において、７０はトラップ検出手段６１からの要求に応じて可能であれば不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段である。図１８はモジュール管理テーブルの構造を示す説明図であり、図において、７１は対応するモジュールが切り離し可能か否かを表す切り離し可能フラグ、７２は対応するモジュールの接続状態、７３は対応するモジュールの切り離し手続き関数アドレスである。なお、その他の構成については上記実施の形態５と同一なのでその重複する説明を省略する。

【００４１】次に動作について説明する。図１９はこの実施の形態６による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートであり、以下、この図１９を参照しながら説明する。トラップ検出手段６１は、まず、ステップＳＴ１９１で、プロセッサ１０からの不当アクセスの通知をトラップとして検出する。次にステップＳＴ１９２で、不当アクセス内容記録手段６２を用いて不当アクセスの内容を二次記憶装置６０上に記録する。そして、ステップＳＴ１９３で、モジュールアンインストール手段７０を用いて可能であれば不当アクセスを要求したモジュールを切り離す。そして、ステップＳＴ１９４で、切り離しが失敗した場合、ステップＳＴ１９５に進み、アクセス要求元に対して不当アクセスを通知する。ステップＳＴ１９４にて、切り離しが成功した場合、そのまま処理を終了する。

【００４２】図２０はモジュールアンインストール処理の流れを示すフローチャートであり、以下、この図２０を参照しながら説明する。モジュールアンインストール手段７０は、まず、ステップＳＴ２０１で、モジュール管理テーブル３０とトラップ検出時のプログラムカウンタから不当アクセスを行ったモジュールを特定する。次にステップＳＴ２０２で、モジュール管理テーブル３０の切り離し可能フラグ７１から特定したモジュールが切り離し可能か否かを検査する。切り離し可能であった場合にステップＳＴ２０３に進み、モジュール管理テーブル３０に登録されている切り離し手続き関数アドレス７３から始まる関数を実行する。そして、ステップＳＴ２０４で、切り離しが成功した場合、ステップＳＴ２０５に進み、アクセス権管理テーブル更新手段４０を用いてモジュール管理テーブル３０内の接続状態７２をアンインストールにする。

【００４３】接続状態７２がアンインストールのモジュール管理テーブル３０のエントリは削除されたものと見なす。そして、ステップＳＴ２０６で、要求元であるトラップ検出手段６１に切り離し成功と通知する。ステップＳＴ２０２にて、切り離しが不可能であった場合はステップＳＴ２０７に進み、要求元であるトラップ検出手段６１に切り離し失敗と通知する。ステップＳＴ２０４にて切り離しが失敗した場合もステップＳＴ２０７に進み、要求元であるトラップ検出手段６１に切り離し失敗と通知する。

【００４４】以上のように、この実施の形態６によれば、モジュール管理テーブル３０に各モジュール毎に切り離しが可能かどうかが定義されており、トラップ検出手段６１がモジュールアンインストール手段７０を用いて、可能であれば不当アクセスを行ったモジュールの切り離しを自動的に行うため、プログラムの実行をできる限り継続させることができる効果がある。

【００４５】

【発明の効果】以上のように、請求項１記載の発明によれば、プロセッサから要求されたアドレスがアクセス権

11

管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段を備えるように構成したので、現在のプログラムカウンタをもとにアクセス権管理テーブルを参照して、モジュール単位で記憶データの保護を行うことができるため、１つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる効果がある。

【００４６】請求項２記載の発明によれば、アクセス権管理テーブルに、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールをモジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えるように構成したので、アクセス権判定手段がアクセスしたデータセグメントについて、アクセス可能モジュールインデックスとして登録されているすべてのモジュールを検査するため、複数のモジュール間で１つのデータセグメントを共有することができる効果がある。

【００４７】請求項３記載の発明によれば、アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えるように構成したので、アクセス権管理テーブルを更新できるのはアクセス権管理テーブル更新手段だけとなるため、アクセス権管理テーブルを誤操作による破壊から守り、安全に更新することができる効果がある。

【００４８】請求項４記載の発明によれば、アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールをデータセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えるように構成したので、モジュールインストール手段がモジュールインストール時にモジュール管理テーブルを作成し、メモリ獲得解放手段がデータセグメントの生成または削除に合わせてデータセグメント管理テーブルの更新を行い、データアクセス宣言手段がモジュールが自発的に宣言した時にアクセス可能モジュールインデックスとして上記モジュールをデータセグメント管理テーブルに登録するため、動的に生成及び削除されるようなデータに対するアクセス権管理テーブルの更新処理を自動的に行うことができる効果がある。

【００４９】請求項５記載の発明によれば、アクセス権

12

判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、トラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えるように構成したので、プロセッサからの不当アクセスの通知を検出したトラップ検出手段が、不当アクセス内容記録手段を用いて不当アクセスの内容を二次記憶装置に記録するため、不当アクセスの内容を確実に且つ容易に知ることができる効果がある。

【００５０】請求項６記載の発明によれば、各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じてフラグを検査し、切り離しが可能であれば切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段とを備えるように構成したので、モジュール管理テーブルに各モジュール毎に切り離しが可能かどうかが定義されており、トラップ検出手段がモジュールアンインストール手段を用いて、可能であれば不当アクセスを行ったモジュールの切り離しを自動的に行うため、プログラムの実行をできる限り継続させることができる効果がある。

【図面の簡単な説明】

【図１】　この発明の実施の形態１による記憶データ保護装置を示すブロック図である。

【図２】　アクセス権管理テーブルの構造を示す説明図である。

【図３】　実施の形態１による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

【図４】　この発明の実施の形態２による記憶データ保護装置を示すブロック図である。

【図５】　モジュール管理テーブルの構造を示す説明図である。

【図６】　データセグメント管理テーブルの構造を示す説明図である。

【図７】　実施の形態２による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

【図８】　この発明の実施の形態３による記憶データ保護装置を示すブロック図である。

【図９】　実施の形態３による記憶データ保護装置におけるアクセス権管理テーブル更新処理の流れを示すフローチャートである。

【図１０】　この発明の実施の形態４による記憶データ保護装置を示すブロック図である。

【図１１】　実施の形態４による記憶データ保護装置におけるモジュールインストール処理の流れを示すフロー

13

チャートである。

【図１２】　メモリ獲得解放処理の流れを示すフローチャートである。

【図１３】　データアクセス宣言処理の流れを示すフローチャートである。

【図１４】　この発明の実施の形態５による記憶データ保護装置を示すブロック図である。

【図１５】　実施の形態５による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートである。

【図１６】　不当アクセス内容記録処理の流れを示すフローチャートである。

【図１７】　この発明の実施の形態６による記憶データ保護装置を示すブロック図である。

【図１８】　モジュール管理テーブルの構造を示す説明図である。

【図１９】　実施の形態６による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートである。

14

【図２０】　モジュールアンインストール処理の流れを示すフローチャートである。

【図２１】　従来の記憶データ保護装置を示すブロック図である。

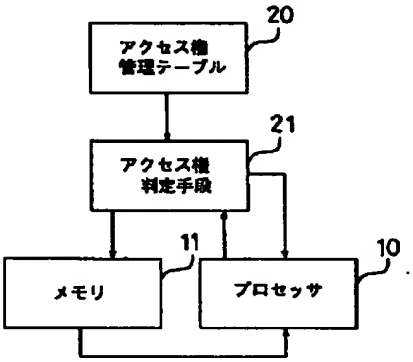
【図２２】　アクセス権管理テーブルの構造を示す説明図である。

【図２３】　記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

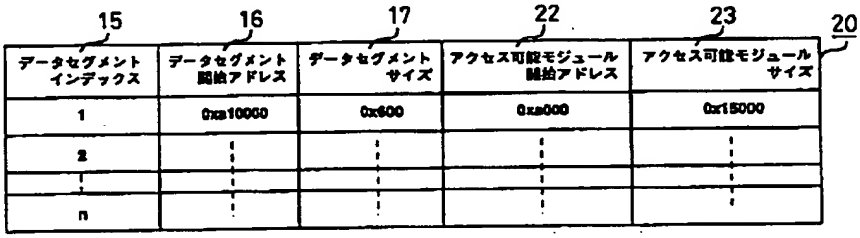
【符号の説明】

１０　プロセッサ、１１　メモリ、２０　アクセス権管理テーブル、２１　アクセス権判定手段、３０　モジュール管理テーブル、３１　データセグメント管理テーブル、４０　アクセス権管理テーブル更新手段、５０　モジュールインストール手段、５１　メモリ獲得解放手段、５２　データアクセス宣言手段、６０二次記憶装置、６１　トラップ検出手段、６２　不当アクセス内容記録手段、７０　モジュールアンインストール手段。

【図１】



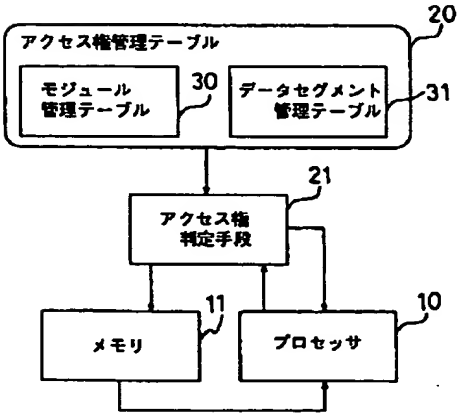
【図２】



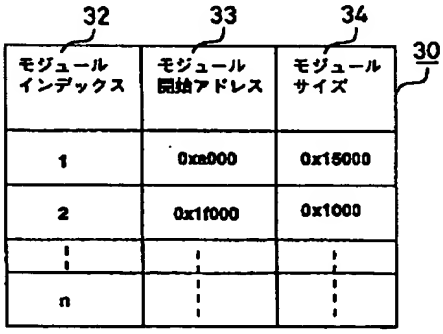
15 データセグメントインデックス	16 データセグメント開始アドレス	17 データセグメントサイズ	22 アクセス可能モジュール開始アドレス	23 アクセス可能モジュールサイズ
1	0xa10000	0x600	0xa000	0x15000
2	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
n	⋮	⋮	⋮	⋮

20

【図４】



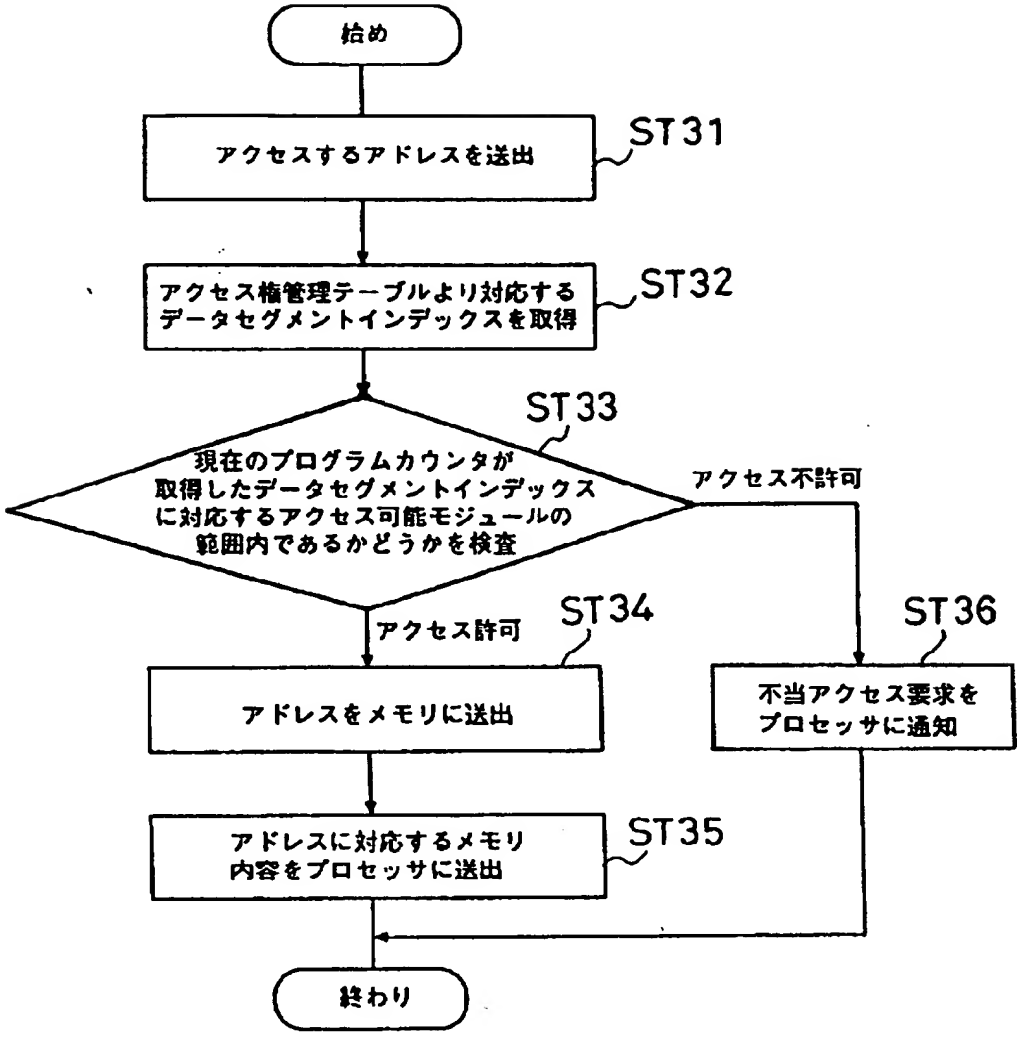
【図５】



32 モジュールインデックス	33 モジュール開始アドレス	34 モジュールサイズ
1	0xa000	0x15000
2	0x1f000	0x1000
⋮	⋮	⋮
n	⋮	⋮

30

【図3】

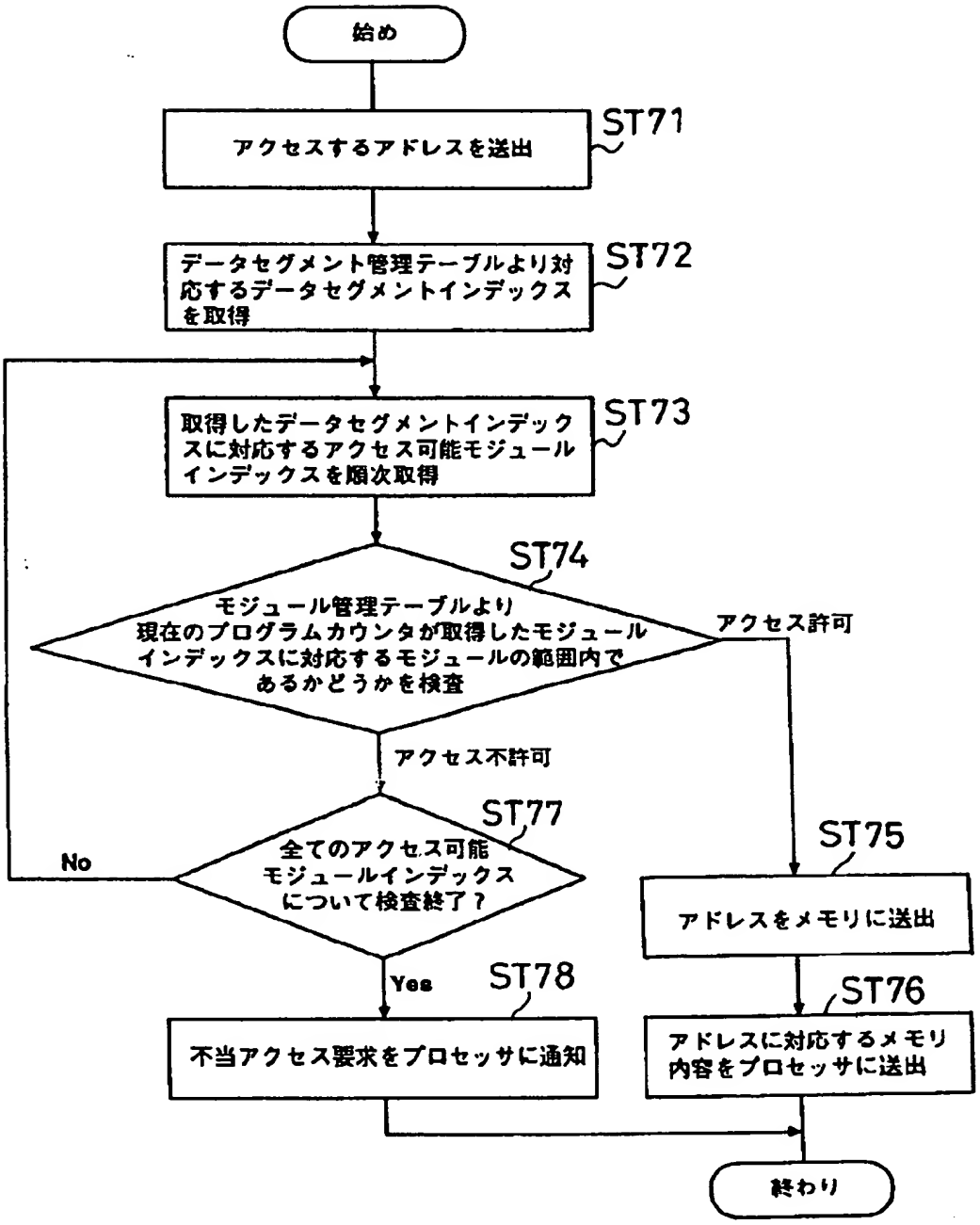


【図6】

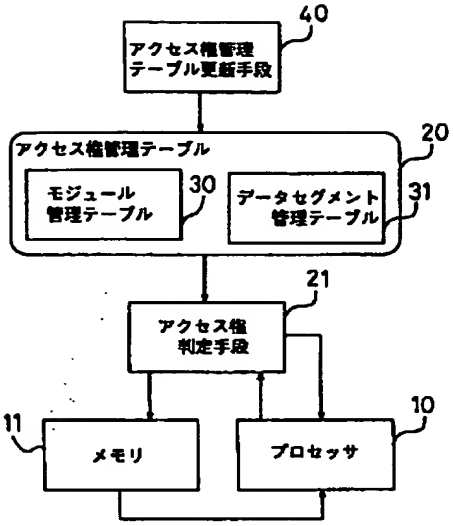
15 データセグメントインデックス	16 データセグメント開始アドレス	17 データセグメントサイズ	35 アクセス可能モジュールインデックス
1	0xa10000	0x500	1, 2, 3, …
2	0xa10500	0x1000	2, 5
⋮	⋮	⋮	⋮
n	⋮	⋮	⋮

31

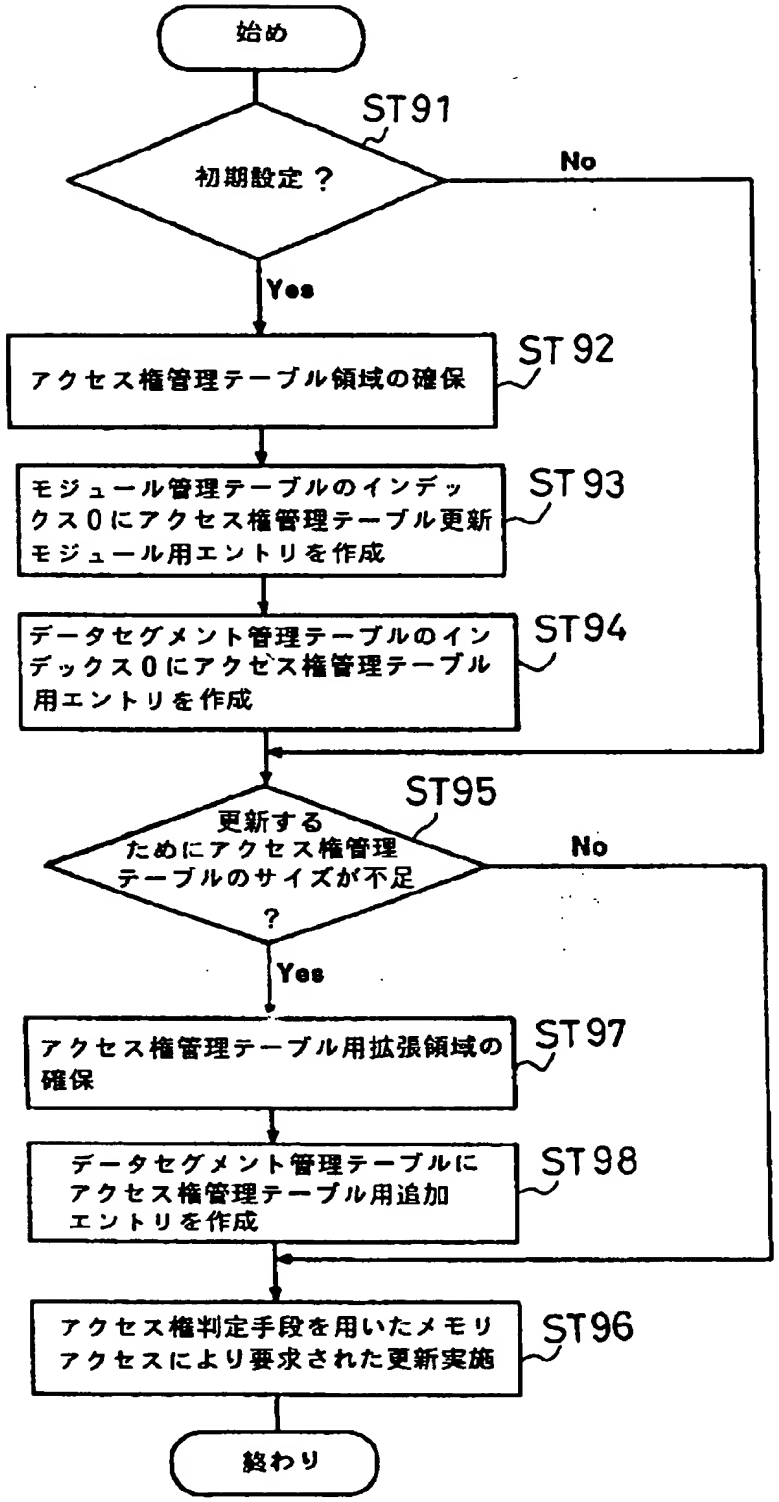
【図7】


始め
アクセスするアドレスを送出
ST71
データセグメント管理テーブルより対応するデータセグメントインデックスを取得
ST72
取得したデータセグメントインデックスに対応するアクセス可能モジュールインデックスを順次取得
ST73
ST74
モジュール管理テーブルより現在のプログラムカウンタが取得したモジュールインデックスに対応するモジュールの範囲内であるかどうかを検査
アクセス許可
アクセス不許可
ST77
全てのアクセス可能モジュールインデックスについて検査終了？
No
Yes
ST75
アドレスをメモリに送出
ST76
アドレスに対応するメモリ内容をプロセッサに送出
ST78
不当アクセス要求をプロセッサに通知
終わり

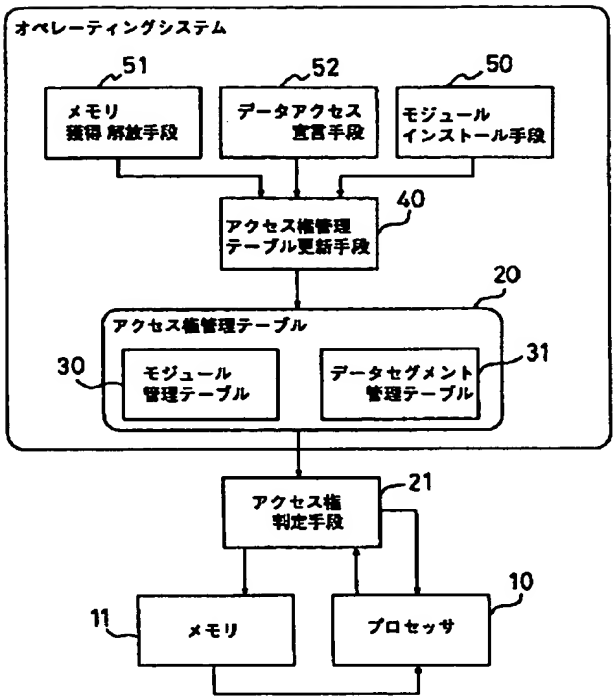
【図８】


40
アクセス権管理
テーブル更新手段
20
アクセス権管理テーブル
モジュール
管理テーブル
30
データセグメント
管理テーブル
31
21
アクセス権
判定手段
11
メモリ
プロセッサ
10

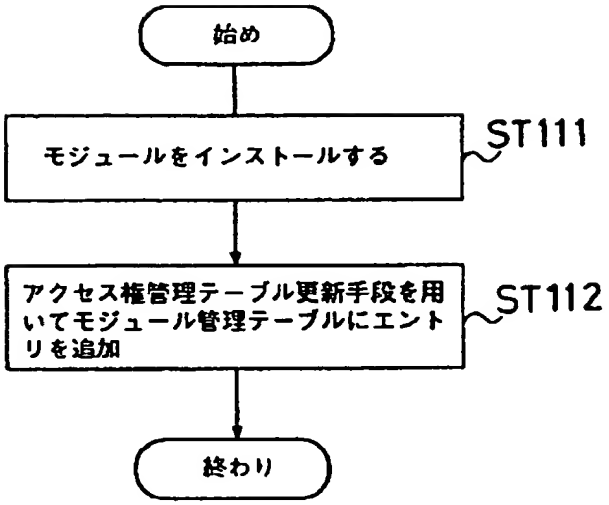
【図９】


始め
ST91
初期設定？
No
Yes
ST92
アクセス権管理テーブル領域の確保
ST93
モジュール管理テーブルのインデックス０にアクセス権管理テーブル更新モジュール用エントリを作成
ST94
データセグメント管理テーブルのインデックス０にアクセス権管理テーブル用エントリを作成
ST95
更新するためにアクセス権管理テーブルのサイズが不足？
No
Yes
ST97
アクセス権管理テーブル用拡張領域の確保
ST98
データセグメント管理テーブルにアクセス権管理テーブル用追加エントリを作成
ST96
アクセス権判定手段を用いたメモリアクセスにより要求された更新実施
終わり

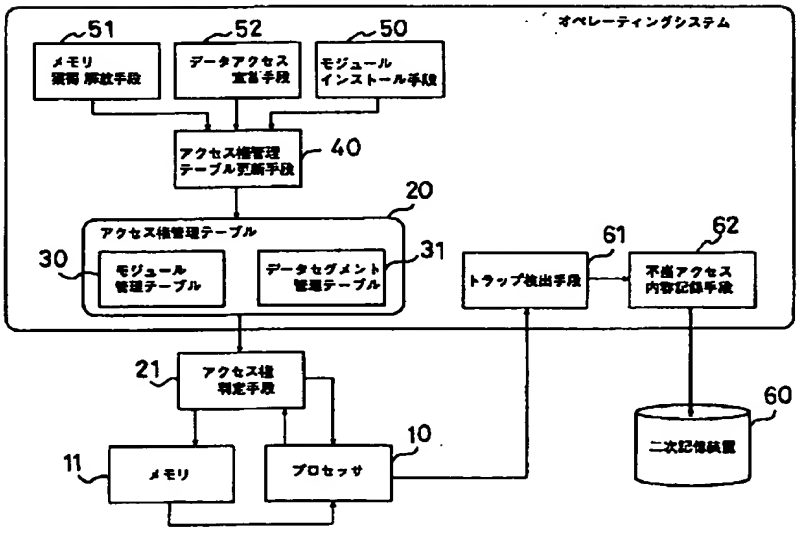
【図１０】


オペレーティングシステム
51
52
50
メモリ
獲得 解放手段
データアクセス
宣言手段
モジュール
インストール手段
40
アクセス権管理
テーブル更新手段
20
アクセス権管理テーブル
30
モジュール
管理テーブル
データセグメント
管理テーブル
31
21
アクセス権
判定手段
10
11
メモリ
プロセッサ

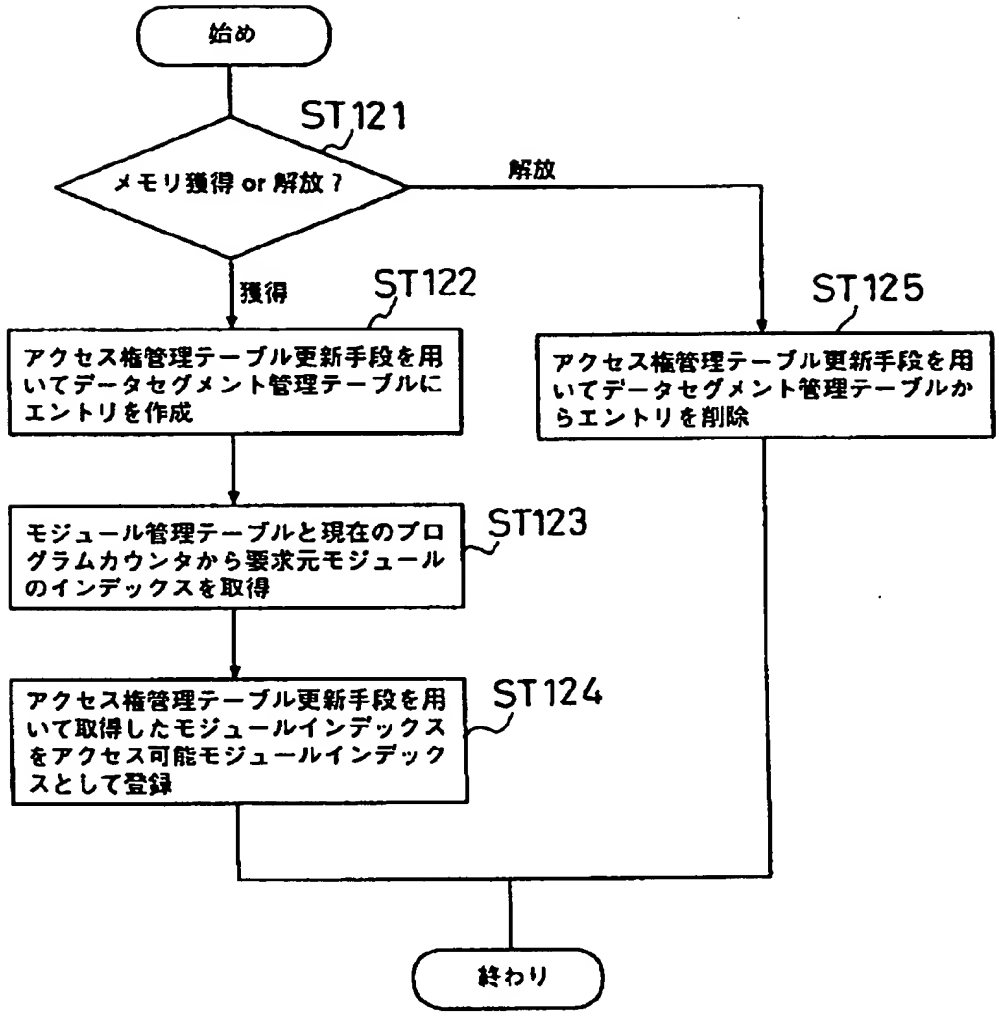
【図１１】


始め
モジュールをインストールする
ST111
アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにエントリを追加
ST112
終わり

【図１４】


オペレーティングシステム
51
52
50
メモリ
獲得 解放手段
データアクセス
宣言手段
モジュール
インストール手段
40
アクセス権管理
テーブル更新手段
20
アクセス権管理テーブル
30
モジュール
管理テーブル
データセグメント
管理テーブル
31
61
62
トラップ検出手段
不当アクセス
内容記録手段
21
アクセス権
判定手段
60
二次記憶装置
11
メモリ
10
プロセッサ

【図１２】

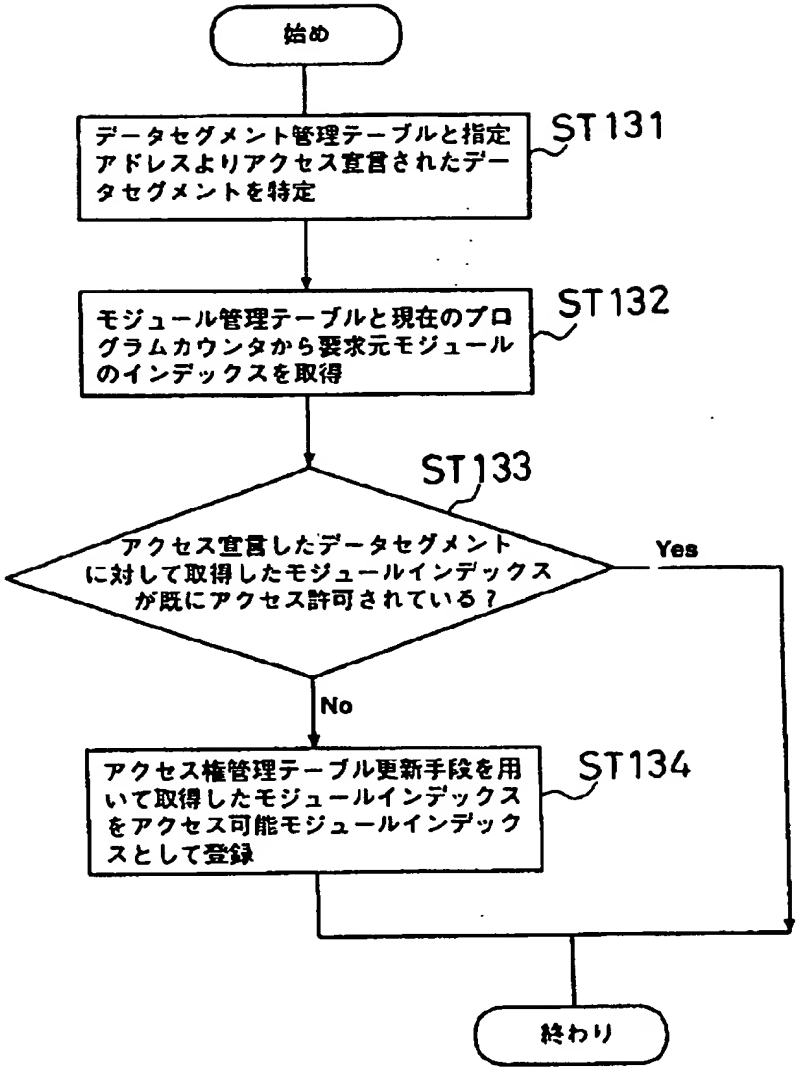


【図１８】

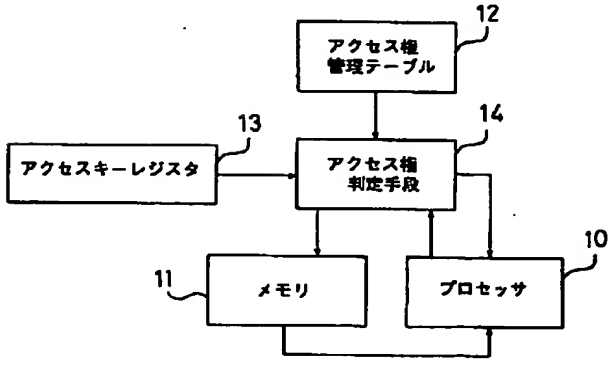
30

32 モジュールインデックス	33 モジュール開始アドレス	34 モジュールサイズ	71 切り離し可能フラグ	72 接続状態	73 切り離し手続き関数アドレス
0：予約 【アクセス権管理テーブル更新モジュール】	0x1dc000	0x2000	×	インストール	――
1	0xa000	0x15000	○	インストール	0x13000
2	0x1f000	0x1000	○	アンインストール	0x1fe00
⋮	⋮	⋮	⋮	⋮	⋮
n	⋮	⋮	⋮	⋮	⋮

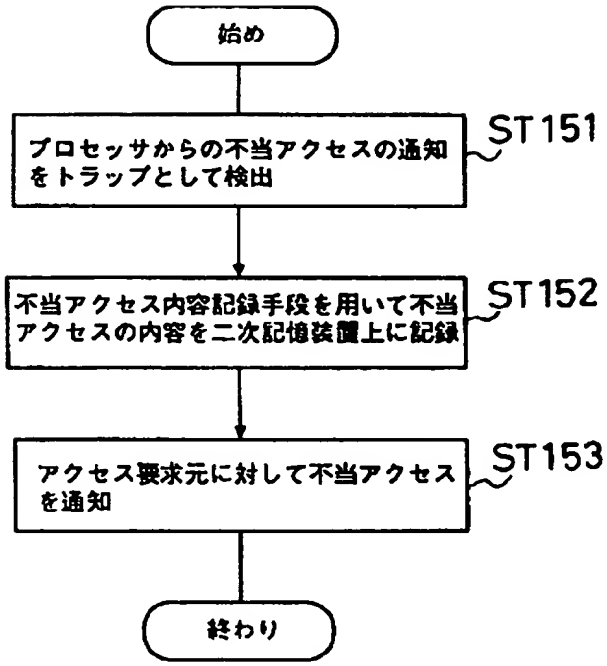
【図１３】


始め
データセグメント管理テーブルと指定アドレスよりアクセス宣言されたデータセグメントを特定
ST131
モジュール管理テーブルと現在のプログラムカウンタから要求元モジュールのインデックスを取得
ST132
ST133
アクセス宣言したデータセグメントに対して取得したモジュールインデックスが既にアクセス許可されている？
Yes
No
アクセス権管理テーブル更新手段を用いて取得したモジュールインデックスをアクセス可能モジュールインデックスとして登録
ST134
終わり

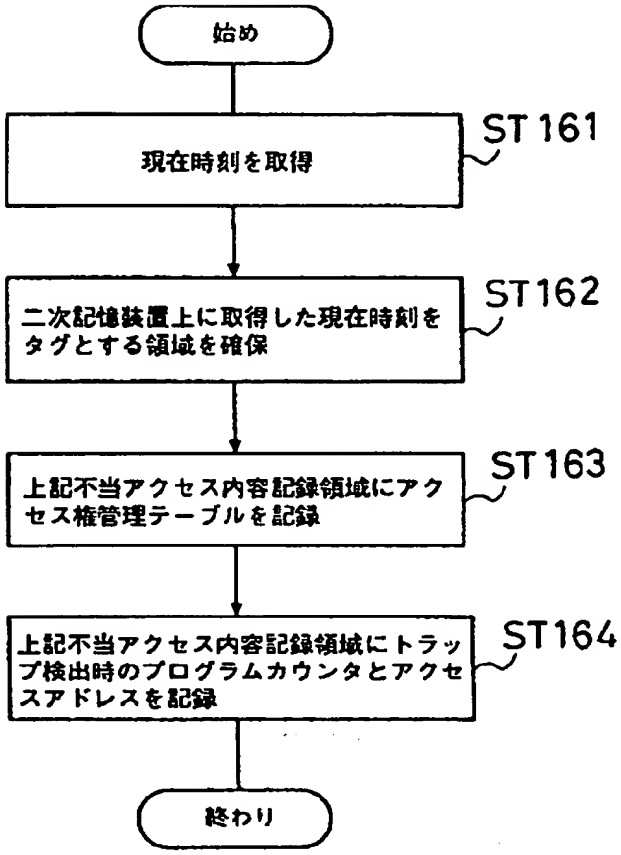
【図２１】


12
アクセス権管理テーブル
13
アクセスキーレジスタ
14
アクセス権判定手段
11
メモリ
10
プロセッサ

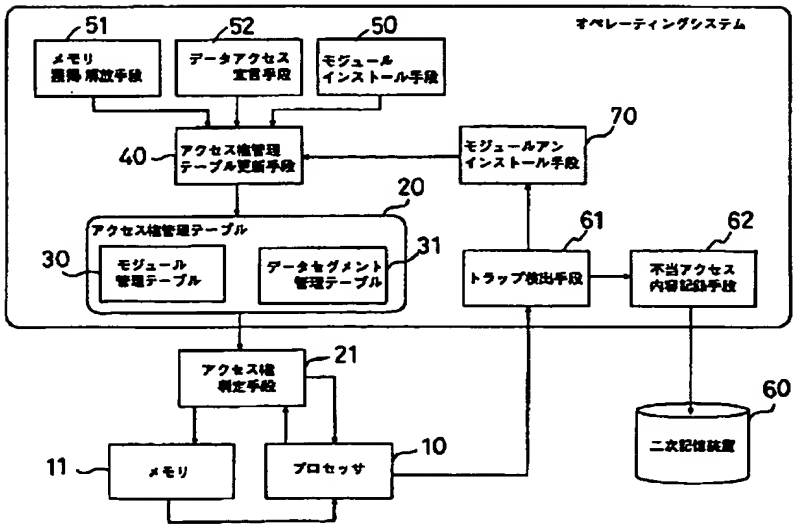
【図１５】


始め
プロセッサからの不当アクセスの通知をトラップとして検出
ST151
不当アクセス内容記録手段を用いて不当アクセスの内容を二次記憶装置上に記録
ST152
アクセス要求元に対して不当アクセスを通知
ST153
終わり

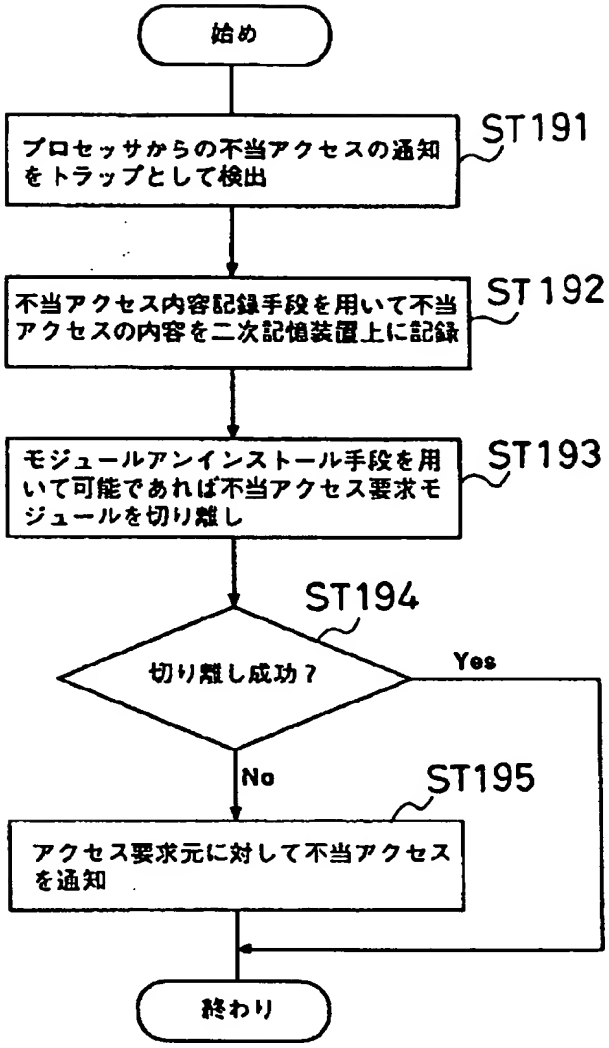
【図１６】


始め
現在時刻を取得
ST161
二次記憶装置上に取得した現在時刻をタグとする領域を確保
ST162
上記不当アクセス内容記録領域にアクセス権管理テーブルを記録
ST163
上記不当アクセス内容記録領域にトラップ検出時のプログラムカウンタとアクセスアドレスを記録
ST164
終わり

【図１７】


オペレーティングシステム
51
52
50
メモリ
データアクセス
モジュール
インストール手段
70
40
アクセス権管理
テーブル更新手段
モジュールアン
インストール手段
20
アクセス権管理テーブル
61
62
30
モジュール
管理テーブル
データセグメント
管理テーブル
31
トラップ検出手段
不当アクセス
内容記録手段
アクセス権
判定手段
21
60
二次記憶装置
11
メモリ
プロセッサ
10

【図１９】

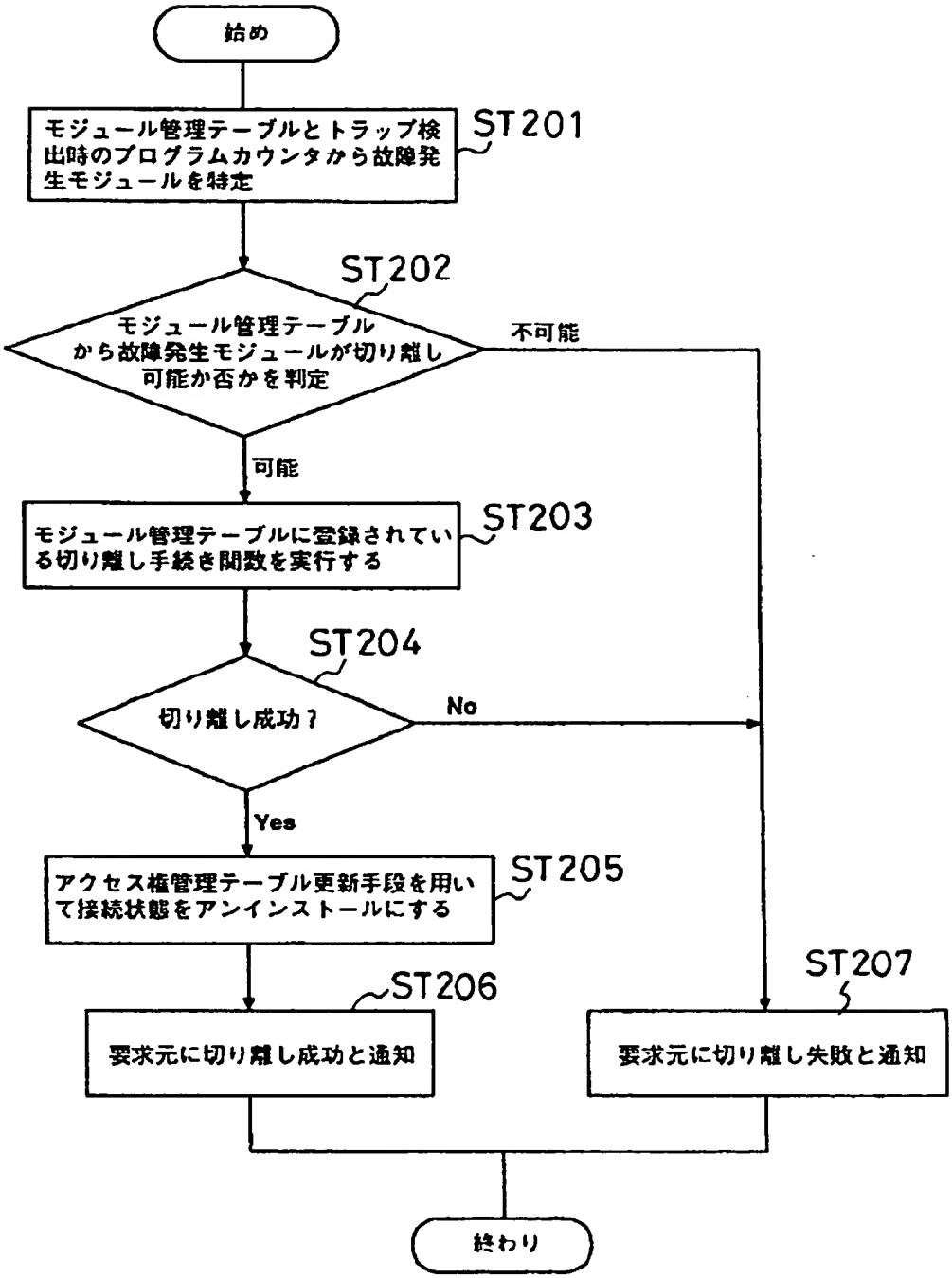


【図２２】

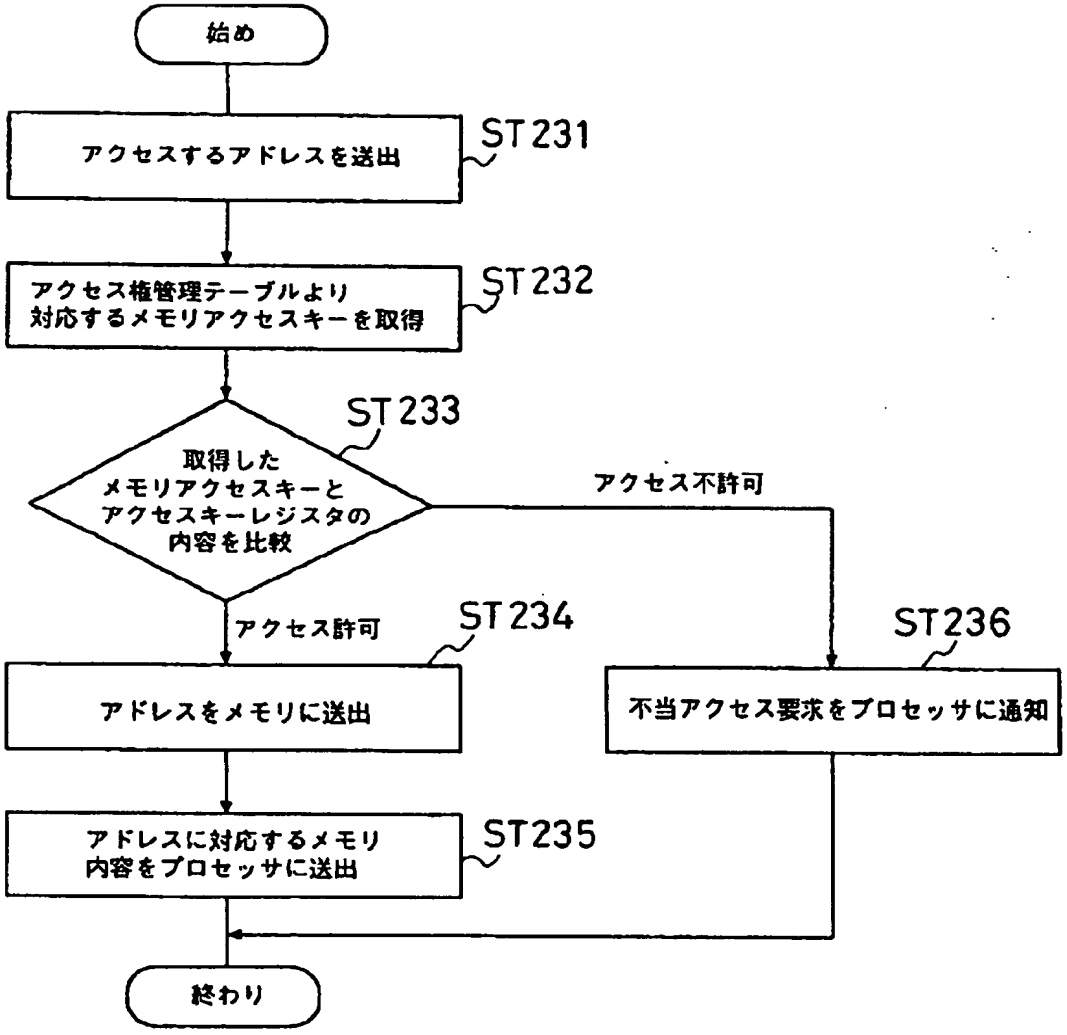
15 データセグメントインデックス	16 データセグメント開始アドレス	17 データセグメントサイズ	18 メモリアクセスキー
1	0xa10000	0x600	32
2	⋮	⋮	⋮
⋮	⋮	⋮	⋮
n	⋮	⋮	⋮

12

【図２０】


始め
モジュール管理テーブルとトラップ検出時のプログラムカウンタから故障発生モジュールを特定
ST201
ST202
モジュール管理テーブルから故障発生モジュールが切り離し可能か否かを判定
不可能
可能
モジュール管理テーブルに登録されている切り離し手続き関数を実行する
ST203
ST204
切り離し成功？
No
Yes
アクセス権管理テーブル更新手段を用いて接続状態をアンインストールにする
ST205
ST206
ST207
要求元に切り離し成功と通知
要求元に切り離し失敗と通知
終わり

【図２３】


始め
アクセスするアドレスを送出
ST231
アクセス権管理テーブルより
対応するメモリアクセスキーを取得
ST232
ST233
取得した
メモリアクセスキーと
アクセスキーレジスタの
内容を比較
アクセス不許可
アクセス許可
ST234
ST236
アドレスをメモリに送出
不当アクセス要求をプロセッサに通知
アドレスに対応するメモリ
内容をプロセッサに送出
ST235
終わり